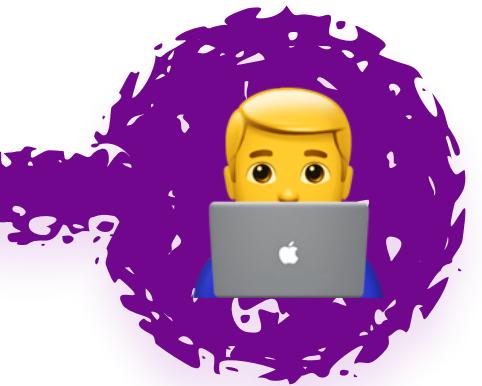


<b>Datas</b>	
<b>28/08/2024</b>	Introdução ao que será o projeto proposto, apresentação do que é a Apple Developer Academy, Introdução a sintaxe da Linguagem C, Tipos de dados primitivos, estruturas condicionais, estruturas de repetição e algoritmos básicos.
<b>04/09/2024</b>	Resolução de exercícios, Espaço para dúvidas e revisão dos conceitos básicos que serão primordiais em aulas posteriores.
<b>11/09/2024</b>	Introdução ao paradigma de programação estruturada, como funciona a linguagem C, ordem de processamento, organização do código, introdução a Strings, Introdução a Struct , Ponteiros, Memória estática e dinâmica, Resolução de exercícios.
<b>18/09/2024</b>	Introdução a estrutura de dados em C , apresentação de array unidimensional, array multidimensional, Algoritmos de ordenação e resolução de exercícios.
<b>25/09/2024</b>	Introdução a Estruturas de dados dinâmicas, Listas encadeadas, conceitos de fila e pilha, Árvores e resolução de exercícios.
<b>02/10/2024</b>	Introdução a Programação orientada a objetos, apresentação dos conceitos de Herança, polimorfismo, encapsulamento e abstração, História da programação orientada a objeto, exemplos de código e resolução de exercícios.
<b>09/10/2024</b>	Introdução a lógica matemática, métodos de raciocínio, tabela verdade, conectivos lógicos, proposições, implicações e equivalência, tautologia, contradição, raciocínio lógico e resolução de exercícios.
<b>16/10/2024</b>	Dicas para entrevista da segunda etapa do processo seletivo do programa Apple Developer Academy.



**Academy**



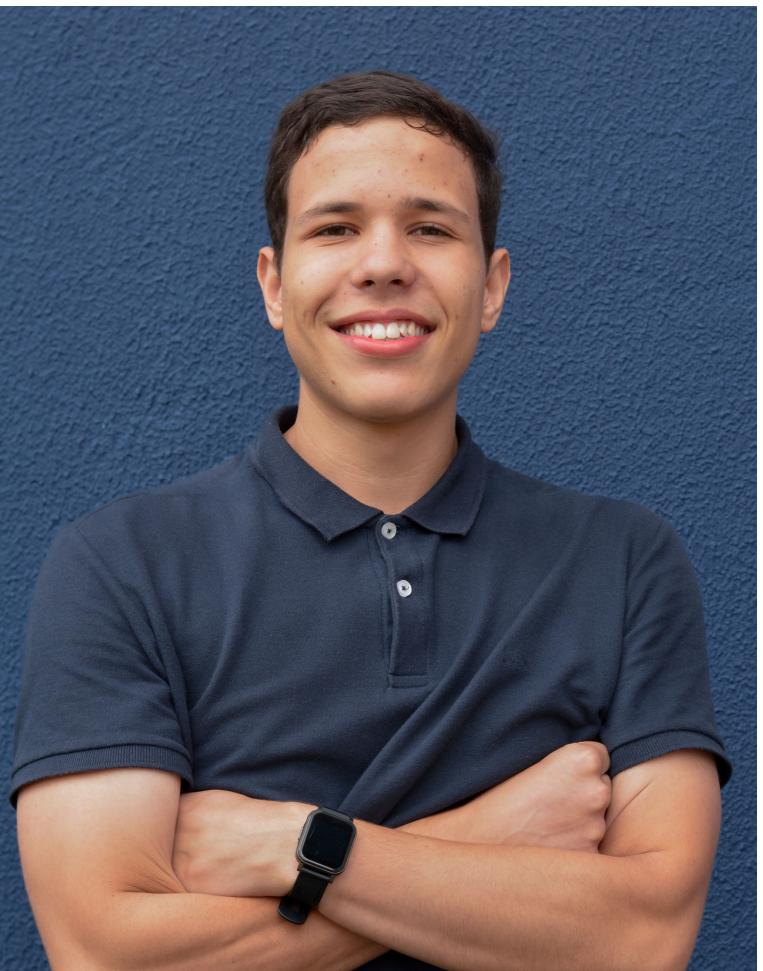
🧐 - Quem somos



**Leonardo Mesquita**



**Luca Gabriel**



**Kauã Miguel**



**João Victor**



- O que é o academy?



 - O que é o academy?

**40 Coders - 10 Designers**

**6 mentores**

# **Nossa Prova - Conteúdos**

# **Sintaxe em C, Ponteiros, Alocação de Memória, Programação Orientada a Objetos, Lógica Matemática**

**Nossa Prova - Conteúdos**

# **Sintaxe em C, Ponteiros, Alocação de Memória, Programação Orientada a Objetos, Lógica Matemática**

**Nossa Prova - Conteúdos**





# Linguagem C

**int    char    float    double**

**int**

4 bytes -2.147.483.648 a 2.147.483.647

char float double

**int**

4 bytes -2.147.483.648 a 2.147.483.647

**char**

1 byte “\$”, “L”, “l” , “5”

**float double**

**int**

4 bytes -2.147.483.648 a 2.147.483.647

**char**

1 byte “\$”, “L”, “I” , “5”

**float**

4 bytes 6 casas decimais

**double**

# **int**

4 bytes -2.147.483.648 a 2.147.483.647

# **char**

1 byte “\$”, “L”, “I” , “5”

# **float**

4 bytes 6 casas decimais

# **double**

8 bytes 15 casas decimais

```
int idade = 21;
```

```
char primeira_letra = 'L' ;
```

```
float saldo_bancario = 25.04;
```

```
double salario_futuro = 15200.654546;
```

**int idades[ ] = { 21, 20 };**

**char nome[ ] = “Leonardo”;**

**float saldos\_bancario[ ] = { 25.04, 500000.50 };**

**double salarios\_futuro[ ] = { 15200.65, 5000.50 };**

**If-else**

**If-else-if**

**Switch**

```
int idade = 21;  
if (idade < 18) {  
    printf("Menor de idade");  
}
```

If-else

If-else-if      Switch

```
int idade = 21;  
if (idade < 18) {  
    printf("Menor de idade"); X  
}
```

If-else

If-else-if      Switch

```
int idade = 21;  
  
if (idade < 18) {  
    printf("Menor de idade");  
}  
else {  
    printf("Maior de idade");  
}
```

If-else

If-else-if      Switch

```
int idade = 21;
```

```
if (idade < 18) {
```

```
    printf("Menor de idade"); X
```

```
} else {
```

```
    printf("Maior de idade"); ✓
```

```
}
```

If-else

If-else-if

Switch

```
int idade = 17;  
  
if (idade < 13) {  
    printf("Criança");  
  
} else if (idade < 18) {  
    printf("Adolescente");  
  
} else {  
    printf("Adulto");  
  
}
```

If-else-if

Switch

```
int idade = 17;  
  
if (idade < 13) {  
  
    printf("Criança"); X  
  
} else if (idade < 18) {  
  
    printf("Adolescente"); ✓  
  
} else {  
  
    printf("Adulto"); X  
  
}
```

If-else-if

Switch

```
int v1 = 20;
```

```
int v2 = 30;
```

```
v1 = (v1 == v2) ? 5 : 10;
```

```
printf("%d", v1);
```

**operador ternário**

```
int v1 = 20;
```

```
int v2 = 30;
```

```
v1 = (v1 == v2) ? 5 : 10;
```

```
printf("%d", v1);
```

**operador ternário**

```
int v1 = 20;
```

```
int v2 = 30;
```

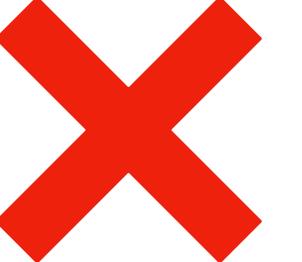
```
v1 = (v1 == v2) ? 5 : 10;
```

```
printf("%d", v1);
```

**operador ternário**

```
int v1 = 20;
```

```
int v2 = 30;
```

```
v1 = (v1 == v2) ? 5 : 10; 
```

```
printf("%d", v1);
```

v1 = 10

operador ternário

```
int opcao = 2;
```

```
switch opcao {
```

```
    case 1:
```

```
        printf("opcao 1");
```

```
        break;
```

```
    case 2:
```

```
        printf("opcao 2");
```

```
        break;
```

```
};
```

## Switch

```
int opcao = 2;  
  
switch opcao {  
  
    case 1:  
  
        printf("opcao 1");  
  
        break;  
  
    case 2:  
  
        printf("opcao 2");  
  
        break;  
  
};
```

opcao 2

Switch

```
int opcao = 2;
```

```
switch opcao {
```

```
    case 1:
```

```
        printf("opcao 1");
```

```
        break; ←
```

```
    case 2:
```

```
        printf("opcao 2");
```

```
        break; ←
```

```
};
```

# Switch

```
int opcao = 1;  
  
switch opcao {  
  
    case 1:  
  
        printf("opcao 1");  
  
    case 2:  
  
        printf("opcao 2");  
  
        break;  
  
};
```

## Switch

```
int opcao = 1;
```

```
switch opcao {
```

```
    case 1:
```

```
        printf("opcao 1");
```

```
    case 2:
```

```
        printf("opcao 2");
```

```
    break;
```

```
};
```

opcao 1

# Switch

```
int opcao = 1;  
  
switch opcao {  
  
    case 1:  
  
        printf("opcao 1");  
  
    case 2:  
  
        printf("opcao 2");  
  
        break;  
};
```

opcao 1  
opcao 2

**Switch**

## **Escolha a opção correta dentre as alternativas para a saída do programa em C.**

```
int num1 = 3, num2 = 2;  
int aux = 5;  
aux = num1;  
num1 = num2;  
  
if (num1 > aux){  
    num1 = 2 * num2;  
} else{  
    num1 = 3 * num2;  
}  
  
num1 = (num1 % 2 ? num1 + 1 : num1);  
num2 = aux;  
  
printf("%d %d %d %d \n", num1, num2, aux, (num1 + num2) );
```

- a) 6 3 3 8**
- b) 4 3 3 7**
- c) 6 2 2 8**
- d) 4 2 2 6**
- e) 6 3 3 9**

## **Escolha a opção correta dentre as alternativas para a saída do programa em C.**

```
int num1 = 3, num2 = 2;  
int aux = 5;  
aux = num1;  
num1 = num2;  
  
if (num1 > aux){  
    num1 = 2 * num2;  
} else{  
    num1 = 3 * num2;  
}  
  
num1 = (num1 % 2 ? num1 + 1 : num1);  
num2 = aux;  
  
printf("%d %d %d %d \n", num1, num2, aux, (num1 + num2) );
```

a) 6 3 3 8

b) 4 3 3 7

c) 6 2 2 8

d) 4 2 2 6

e) 6 3 3 9

**num1 = 3 , num2 = 2 , aux = ?**

→ int num1 = 3, num2 = 2;  
int aux = 5;  
aux = num1;  
num1 = num2;

if (num1 > aux){  
 num1 = 2 \* num2;  
} else{  
 num1 = 3 \* num2;  
}

**num1 = 3 , num2 = 2 , aux = 5**

→ int aux = 5;  
aux = num1;  
num1 = num2;

if (num1 > aux){  
 num1 = 2 \* num2;  
} else{  
 num1 = 3 \* num2;  
}

**num1 = 3 , num2 = 2 , aux = 3**

→ aux = num1;

num1 = num2;

if (num1 > aux){

    num1 = 2 \* num2;

} else{

    num1 = 3 \* num2;

}

num1 = (num1 % 2 ? num1 + 1 : num1);

**num1 = 2 , num2 = 2 , aux = 3**

→ num1 = num2;

```
if (num1 > aux){  
    num1 = 2 * num2;  
} else{  
    num1 = 3 * num2;  
}
```

```
num1 = (num1 % 2 ? num1 + 1 : num1);  
num2 = aux;
```

**num1 = 2 , num2 = 2 , aux = 3**

→ if (num1 > aux){ **False**  
    num1 = 2 \* num2;  
} else{  
    num1 = 3 \* num2;  
}

num1 = (num1 % 2 ? num1 + 1 : num1);  
num2 = aux;

printf("%d %d %d %d \n", num1, num2, aux, (num1 +

**num1 = 6 , num2 = 2 , aux = 3**

→ num1 = 3 \* num2;  
}

num1 = (num1 % 2 ? num1 + 1 : num1);  
num2 = aux;

printf("%d %d %d %d \n", num1, num2, aux, (num1 + num2));

**num1 = 6 , num2 = 2 , aux = 3**

→ num1 = (num1 % 2 ? num1 + 1 : num1);  
num2 = aux; 1 0

printf("%d %d %d %d \n", num1, num2, aux, (num1 + num2));

**num1 = 6 , num2 = 3 , aux = 3**

→ num2 = aux;

printf("%d %d %d %d \n", num1, num2, aux, (num1 + num2));

**num1 = 6 , num2 = 3 , aux = 3**

→ printf(“%d %d %d %d \n”, num1, num2, aux, (num1 + num2) );

**Console: 6 3 3 9**

## Escolha a opção correta dentre as alternativas para a saída do programa em C.

```
int num1 = 3;
```

```
switch (num1) {  
    case 1:  
        num1 *= 3;  
        break;  
    case 2:  
        num1 /= 3;  
        break;  
    case 3:  
        num1 += 4;  
    default:  
        num1 = 0;  
        break;  
}
```

```
printf("%d \n", num1);
```

- a) 9
- b) 7
- c) 1
- d) 0
- e) 3

## Escolha a opção correta dentre as alternativas para a saída do programa em C.

```
int num1 = 3;
```

```
switch (num1) {  
    case 1:  
        num1 *= 3;  
        break;  
    case 2:  
        num1 /= 3;  
        break;  
    case 3:  
        num1 += 4;  
    default:  
        num1 = 0;  
        break;  
}
```

```
printf("%d \n", num1);
```

- a) 9
- b) 7
- c) 1
- d) 0
- e) 3

**num1 = 3**

→ case 1:

```
    num1 *= 3;  
    break;
```

case 2:

```
    num1 /= 3;  
    break;
```

case 3:

```
    num1 += 4;
```

default:

```
    num1 = 0;  
    break;
```

**Não conforma**

```
num1 = 3
```

→ case 2:

```
    num1 /= 3;
```

```
    break;
```

case 3:

```
    num1 += 4;
```

default:

```
    num1 = 0;
```

```
    break;
```

```
}
```

**Não conforma**

```
printf("%d \n", num1);
```

**num1 = 7**

→ case 3:

    num1 += 4;

default:

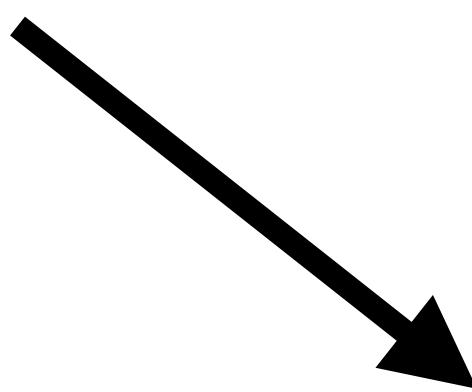
    num1 = 0;

    break;

}

printf("%d \n", num1);

**Conforma**



    num1 = num1 + 4;

```
num1 = 0
```

→ default:

```
    num1 = 0;
```

```
    break;
```

```
}
```

```
printf("%d \n", num1);
```

**Conforma**

**num1 = 0**

→ printf("%d \n", num1);

**Console: 0**

**For**      **While**      **Do-While**

```
int num = 0;
```

```
int i;
```

```
for (i = 0; i < 5; i++) {
```

```
    num += 1;
```

```
}
```

**For**

**While      Do-While**

```
int num = 0;
```

```
int i;
```

```
for (i = 0; i < 5; i++) {
```

```
    num += 1;
```

```
}
```

For

While      Do-While

```
int num = 0;
```

```
int i;
```

```
for (i = 0; i < 5; i++) {
```

```
    num += 1;
```

```
}
```

**For**

**While      Do-While**

```
int num = 0;
```

```
int i;
```

```
for (i = 0; i < 5; i++) {
```

```
    num += 1;
```

```
}
```

**For**

**While      Do-While**

```
int num = 0;
```

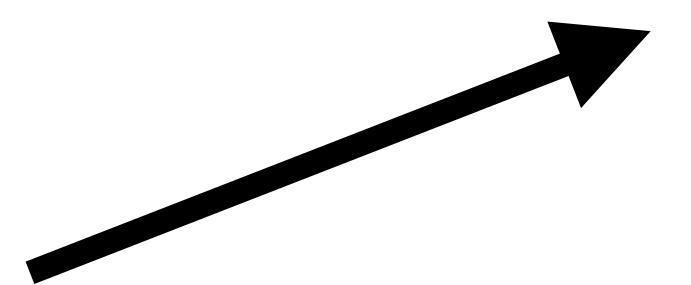
```
int i;
```

```
for (i = 0; i < 5; i++) {
```

```
    num += 1;
```

```
}
```

$i = i + 1$



**For**

**While**

**Do-While**

```
int num = 0;
```

```
int i;
```

```
for (i = 0; i < 5; i++) {
```

```
    num += 1;           i = 1
```

```
}                 num += 1;           i = 2
```

```
                 num += 1;           i = 3
```

```
                 num += 1;           i = 4
```

```
For                 num += 1;           i = 5
```

While      Do-While

```
int num = 0;
```

```
int i;
```

```
for (i = 0; i < 5; i++) {
```

```
    num += 1;
```

```
}
```

```
num = 5
```

For

While      Do-While

```
int num = 0;
```

```
while (num < 5) {
```

```
    num++;
```

```
}
```

**While**

**Do-While**

```
int num = 0;
```

```
while (num < 5) {
```

```
    num++;
```

**num = 1**

```
}
```

```
    num++;
```

**num = 2**

```
    num++;
```

**num = 3**

```
    num++;
```

**num = 4**

```
    num++;
```

**num = 5**

**While**

**Do-While**

```
int num = 0;
```

```
while (num < 5) {
```

```
    num++;
```

```
}
```

```
num = 5
```

**While**

**Do-While**

```
int num = 5;  
do {  
    num++;  
} while (num < 5);
```

## Do-While

```
int num = 5;  
do {  
    num++;  
} while (num < 5);
```

num = 6

**Do-While**

**Assinale a opção CORRETA sobre o número de vezes que a palavra "Teste" será mostrada, ao executarmos o programa:**

```
int i, j;  
  
for(i = 0; i < 2; i++) {  
    for(j = 1; j < 3; j++) {  
        printf("\nTeste");  
    }  
}
```

a) 3  
b) 4  
c) 5  
d) 6  
e) 7

**Assinale a opção CORRETA sobre o número de vezes que a palavra "Teste" será mostrada, ao executarmos o programa:**

```
int i, j;  
  
for(i = 0; i < 2; i++) {  
    for(j = 1; j < 3; j++) {  
        printf("\nTeste");  
    }  
}
```

- a) 3
- b) 4
- c) 5
- d) 6
- e) 7

## **Escolha a opção correta dentre as alternativas para a saída do programa em C.**

```
int vetor[ ] = {1,2,4,5,7,9,10};  
int quant_maxima = 3;  
int quant = 0;  
int soma = 0;  
int i = 0;  
  
do {  
    if (vetor[ i ] % 2 != 0)  
        quant++;  
    else  
        soma += vetor[ i ];  
    i++;  
} while (i < 7 && quant_maxima != quant);  
  
printf("%d \n", soma);
```

- a) 6**
- b) 12**
- c) 7**
- d) 3**
- e) 5**

## Escolha a opção correta dentre as alternativas para a saída do programa em C.

```
int vetor[ ] = {1,2,4,5,7,9,10};  
int quant_maxima = 3;  
int quant = 0;  
int soma = 0;  
int i = 0;  
  
do {  
    if (vetor[ i ] % 2 != 0)  
        quant++;  
    else  
        soma += vetor[ i ];  
    i++;  
} while (i < 7 && quant_maxima != quant);  
  
printf("%d \n", soma);
```

- a) 6
- b) 12
- c) 7
- d) 3
- e) 5



**Linkedin**

Leonardo Mesquita Alves

